REMARKS

Claims 54-75 are pending in the present application. The Office Action and cited references have been considered. Favorable reconsideration and such allowance are respectfully urged.

New claims 54 - 75 are presented for reconsideration. Claims 1 - 53 have been canceled.

References to paragraph numbers in the Specification are taken from the Official Publication hereof, Publication No. 2003/0130813.

REJECTIONS UNDER 35 U.S.C. §103

First Rejection.

Claims 1, 3-6, 8-10, 21-23 and 25-27 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Shrote, U.S. Patent No. 5,774,358 in view of Katz *et al.*, U.S. Patent No. 4,493,027. These claims have been canceled, and replaced by new claims 54 - 70.

Shrote discloses a test generator system in which an instruction/data stream is used to verify a hardware design. The instruction stream is produced in part by generation of exception events. The exception event possibilities are known to the system, and are described in an event behavior rule

table (Example 2, Cols. 7 - 8). Shrote discloses a functional block 204 that receives the event behavior rules, and compiles them into a format that may be used to generate specific instruction to investigate the events. The events are explicitly generated (Col. 9, lines 5 - 10), using a code/data generation entity 210, which generates a code/data thread for each exception. The threads can then be run in succession (Col. 11, lines 30 - 40). The event behavior rule table specifies a single combination and sequence to be used for investigation of the event (last 17 lines of table in Example 2; Col. 9, lines 3-10).

The claimed invention operates in a different manner from that of Shrote. Rather than explicitly generating events in order to investigate their effect on the system-under-test, a primary input stream of partially specified instructions is generated until triggering conditions of an event are recognized. The test generator then reacts to the triggering condition either by switching to an alternate input stream or by continuing the primary input stream for generation of subsequent program instructions (e.g., Specification, Para [0132]). The alternate input stream has been previously prepared and embedded into a data structure representing the current event. These features are recited in the language of the new independent claims 54, 60, 66 and 71. The alternate

input stream can be selected or ignored according to a governing policy in order to investigate desired aspects of the system-under-test when the current event occurs.

The advantage of operation in this manner is that events that cannot be explicitly generated can still be investigated. Indeed, as a system-under-test becomes increasingly complex, such events become more frequent. For example, timing issues and race conditions among different resources of the system-under-test can unexpectedly produce contentions. Even though such "events" can be devastating to the operation of the system-under-test, they often cannot be readily replicated or explicitly generated, due to random factors inherent in the design or due to the algorithmic complexity involved in explicitly generating some types of events. Nevertheless, verification of the system-under-test when such events occur is well-handled using the invention (Specification, para. [0063]). However, such unexpected events cannot be handled by the Shrote system. Nor does Shrote suggest any modification that would allow handling of unexpected events. Furthermore, the Shrote system is much more limited, as only one combination and sequence is possible to be generated at a time in the input stream using the event behavior rule table. The inventive technique is more flexible, in that either the primary input stream or an

alternate input stream may be used for generation of subsequent program instructions. New dependent claims 57, 63, 69 and 74 recite a nonpredetermined occurrence, which is an unexpected, occurrence, of triggering conditions of events in the input stream. Such unexpected event conditions are not explicitly created by a user.

New dependent claims 58, 64, 70 and 75 recite processing a set of events in order of priority. The Examiner has asserted that Katz *et al.* disclose processing of events in order of priority. Applicant submits that this feature is not shown to have been employed in a test generator as claimed herein. Katz *et al.* disclose a microcode control system for a digital computer. Applicant urges that one seeking to improve a test generator would not look to the microcode literature for its teachings on event processing, as this is a specialized art which is non-analogous to the subject matter of the present invention.

Second Rejection.

Claims 11 and 28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Shrote, U.S. Patent No. 5,774,358 in view of Katz *et al.*, U.S. Patent No. 4,493,027. Claims 11 and 28 have been canceled.

Third Rejection.

Claims 12 - 20 and 29 - 53 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Shrote, U.S. Patent No. 5,774,358 in view of Katz *et al.*, U.S. Patent No. 4,493,027 and further in view of Matsuba *et al.*, U.S. Patent No. 6,467,078. Matsuba *et al.* disclose a development system that is adapted to develop software programs for a real time control system, based on a state transition matrix, in which actions are taken based on an intersection of an event and a cell in the matrix (Col. 11, line 66 through Col. 12, line 10). Events are input into the system using a script editor (Col. 18 lines 60 - 65). A simulator includes an event analysis section, which is capable of rearranging events of a script file into a sequence of events. (Col. 19, lines 27 - 35). It appears that the events must be explicitly included in the script file. The system of Matsuba *et al.* includes a mechanism for simulating and testing the soft ware developed by the system. However, while it allows the user to define events in a script and to verify that the target software reacts correctly, these events must be explicitly introduced by the user. Furthermore, the system does not provide for alternate input streams upon occurrence of events, as recited in new independent apparatus claims 66 and 71. As in the system of Katz *et al.*, discussed above, the system of Matsuba

et al. is capable of dealing only with events that the user is able to generate in the script file.

Concluding Matters

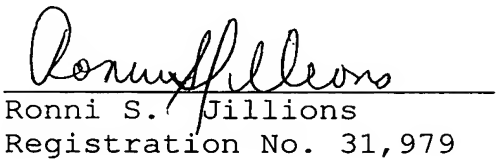
It is believed that the amendments and remarks presented hereinabove are fully responsive to all the grounds of rejection and objections raised by the Examiner, and that the Application is now in order for allowance.

Applicant submits that the application is in condition for allowance and early notice to this effect is most earnestly solicited.

If the Examiner has any questions, he is invited to call the undersigned at (202) 628-5197.

Respectfully submitted,

BROWDY AND NEIMARK, P.L.L.C.
Attorneys for Applicant(s)

By 
Ronni S. Jillions
Registration No. 31,979

RSJ:ma
Telephone No.: (202) 628-5197
Facsimile No.: (202) 737-3528
G:\bn\c\colb\adir1\pto\Amendment-B.doc